**## Assignment Part-1**

Q1. Why do we call Python as a general purpose and high-level programming language?

Python ia programming language that can write in a human readable form.

Q2. Why is Python called a dynamically typed language?

In Python, we don’t need to declare the type of variable. It stores the value of a variable to some memory location and then binds that variable name to that memory container. And makes the contents accessible through that variable name. It also takes care of memory managemant

Q3. List some pros and cons of Python programming language?

|  |  |
| --- | --- |
| **Pros** | **Cons** |
| Easy to Read, Learn and Write | Slow Speed |
| Improved Productivity | Not Memory Efficient |
| Interpreted Language | Weak in Mobile Computing |
| Dynamically Typed | Database Access |
| Free and Open-Source | Runtime Errors |
| Vast Libraries Support |  |
| Portability |  |

Q4. In what all domains can we use Python?

1. Web development

2. Data Science

3. OS development

4. Scientific programming

5. Gaming

Q5. What are variable and how can we declare them?

Variable is a nmae given to a specific memory location.

We don’t declare variables, simply assign them.

Q6. How can we take an input from the user in Python?

By the keyword input()

var = input("Enter name: ")

Q7. What is the default datatype of the value that has been taken as an input using input() function?

By default, the input() function will convert all the information it receives into a string.

Q8. What is type casting?

The conversion of one data type into the other data type is known as type casting in python or type conversion in python. Python supports a wide variety of functions or methods like: int(), float(), str(), ord(), hex(), oct(), tuple(), set(), list(),dict(), etc. for the type casting in python.

There are two varieties of typecasting in python namely -

**Explicit Conversion**(Explicit type casting in python), and

(Implicit type casting in python).

The conversi**Implicit Conversion**on of one data type into another, done via user intervention or manually as per the requirement, is known as explicit type conversion. In Implicit type conversion, Python automatically converts one data type to another data type without any user's need.

Q9. Can we take more than one input from the user using single input() function? If yes, how? If no, why?

Using split() Method

The split() method is useful for getting multiple inputs from users. The syntax is given below.

input().split(separator,maxsplit)

separator: This is a delimiter. The string splits at this specified separator. If is not provided, then any white space is a separator.

maxsplit: It is a number, which tells us to split the string into a maximum of provided number of times. If it is not provided then the default is -1 which means there is no limit.

Eg:

x,y,z = input("Enter variables: ").split(",",3)

print(x,y,z)

Enter variables: how,are,you

how are you

Q10. What are keywords?

Python keywords are reserved words. They are used by python interpreters to understand the program. Keywords define the structure of programs.

Value Keywords: True, False, None.

Operator Keywords: and, or, not, in, is.

Control Flow Keywords: if, elif, else.

Iteration Keywords: for, while, break, continue, else.

Structure Keywords: def, class, with, as, pass, lambda.

Returning Keywords: return, yield.

Import Keywords: import, from, as.

Q11. Can we use keywords as a variable? Support your answer with reason.

No, Keywords in Python are reserved words that can not be used as a variable name, unction name, or any other identifier.

Q12. What is indentation? What's the use of indentaion in Python?

Python indentation is a way of telling the Python interpreter that a series of statements belong to a particular block of code. Python uses space/tab as indentation to indicate the same to the compiler.All the statements with the same distance (space) to the right, belong to the same block.

Q13. How can we throw some output in Python?

Python uses the print() function to output data.

Q14. What are operators in Python?

Python Operator falls into 7 categories:

Python Arithmetic Operator

Python Relational Operator

Python Assignment Operator

Python Logical Operator

Python Membership Operator

Python Identity Operator

Python Bitwise Operator

Q15. What is difference between / and // operators?

Normal division (/) returns a fractional number, whereas floor division (//) truncates the decimal part and returns the quotient.

Q16. Write a code that gives following as an output.

```

iNeuroniNeuroniNeuroniNeuron

```

print(“iNeuroniNeuroniNeuroniNeuron”)

Q17. Write a code to take a number as an input from the user and check if the number is odd or even.

num = input(“Enter a number”)

if num%2 ==0:

print(“Number is even”)

else:

print(“Number is odd”)

Q18. What are boolean operator?

The logical operators and, or and not are also referred to as boolean operators.

The Python Boolean type has only two possible values: True and False.

The type() of both False and True is bool. The type bool is built in, meaning it’s always available in Python and doesn’t need to be imported.

Q19. What will the output of the following?

```

1 or 0 - True or False

0 and 0 – False and False

True and False and True - True and False

1 or 0 or 0 - True or False

```

Q20. What are conditional statements in Python?

Python has 3 key Conditional Statements:

if statement

if-else statement

if-elif-else ladder

Q21. What is use of 'if', 'elif' and 'else' keywords?

if…elif…else are conditional statements that provide you with the decision making that is required when you want to execute code based on a particular condition.

The if…elif…else statement helps automate that decision making process.

The if condition is considered the simplest of the three and makes a decision based on whether the condition is true or not. If the condition is true, it prints out the indented expression. If the condition is false, it skips printing the indented expression.

The if-else condition adds an additional step in the decision-making process compared to the simple if statement. The beginning of an if-else statement operates similar to a simple if statement; however, if the condition is false, instead of printing nothing,the indented expression under else will be printed.

The most complex of these conditions is the if-elif-else condition. When you run into a situation where you have several conditions, you can place as many elif conditions as necessary between the if condition and the else condition.

Q22. Write a code to take the age of person as an input and if age >= 18 display "I can vote". If age is < 18 display "I can't vote".

age= input(“Enter age of a person”)

if age>18:

print(“I can vote”)

elif age<18:

print(“I can’t vote”)

Q23. Write a code that displays the sum of all the even numbers from the given list.

```

numbers = [12, 75, 150, 180, 145, 525, 50]

```

numbers = [12, 75, 150, 180, 145, 525, 50]

sum=0

for i in numbers:

if i%2==0:

sum= sum+i

print("Sum of numbers ", sum)

Q24. Write a code to take 3 numbers as an input from the user and display the greatest no as output.

num1,num2,num3 = input("Enter three numbers: ").split(",",3)

if (num1 >= num2) and (num1 >= num3):

largest = num1

elif (num2 >= num1) and (num2 >= num3):

largest = num2

else:

largest = num3

print("The largest number is", largest)

Q25. Write a program to display only those numbers from a list that satisfy the following conditions

- The number must be divisible by five

- If the number is greater than 150, then skip it and move to the next number

- If the number is greater than 500, then stop the loop

```

numbers = [12, 75, 150, 180, 145, 525, 50]

```

numbers = [12, 75, 150, 180, 145, 525, 50]

for i in numbers:

if i > 500:

break

elif i > 150:

continue

# check if number is divisible by 5

elif i % 5 == 0:

print(i)

Q26. What is a string? How can we declare string in Python?

Strings can be created by enclosing characters inside a single quote or d ouble-quotes. Even triple quotes can be used in Python but generally used to represent multiline strings and docstrings.

str1 = 'Ineuron'

print('str1 = ', str1)

Q27. How can we access the string using its index?

Indexing allows you to access individual characters in a string directly by using a numeric value.

str\_index = "Ineuron!"

print(str\_index[0])

Q28. Write a code to get the desired output of the following

```

string = "Big Data iNeuron"

desired\_output = "iNeuron"

```

print("desired\_output = ", string[8:])

Q29. Write a code to get the desired output of the following

```

string = "Big Data iNeuron"

desired\_output = "norueNi"

```

print("desired\_output = ", string[-1:8:-1])

Q30. Resverse the string given in the above question.

print("desired\_output = ", string[-1::-1])

Q31. How can you delete entire string at once?

del string

Q32. What is escape sequence?

Escape characters can be classified as non-printable characters when backslash precedes them. The print statements do not print escape characters.

|  |  |
| --- | --- |
| **Code** | **Description** |
| \’ | Single quotation |
| \\ | Backslash |
| \n | New Line |
| \r | Carriage Return |
| \t | Tab |
| \b | Backspace |
| \f | Form feed |
| \ooo | Octal equivalent |
| \vhhh | Hexadecimal equivalent |

Q33. How can you print the below string?

```

'iNeuron's Big Data Course'

```

str2=”iNeuron's Big Data Course”

print(str2)

Q34. What is a list in Python?

List stores heterogenous kind of data in contiguous memory location.

Q35. How can you create a list in Python?

A list is created by placing elements inside square brackets [], separated by commas.

Q36. How can we access the elements in a list?

To access elements in lists, we can use the square brackets for slicing along with the index or indices to obtain value available at that index.

Q37. Write a code to access the word "iNeuron" from the given list.

```

lst = [1,2,3,"Hi",[45,54, "iNeuron"], "Big Data"]

```

print(lst[4][2])

Q38. Take a list as an input from the user and find the length of the list.

lst = [1,2,3,"Hi",[45,54, "iNeuron"], "Big Data"]

lst1= lst[4][2]

print(len(lst1))

Q39. Add the word "Big" in the 3rd index of the given list.

```

lst = ["Welcome", "to", "Data", "course"]

```

lst[3]="Big"

print(lst)

Q40. What is a tuple? How is it different from list?

Tuple is also similar to list but contains immutable objects. Tuple processing is faster than List.

|  |  |
| --- | --- |
| **List** | **Tuple** |
| Lists are mutable | Tuples are immutable |
| The implication of iterations is Time-consuming | The implication of iterations is comparatively Faster |
| The list is better for performing operations, such as insertion and deletion. | Tuple data type is appropriate for accessing the elements |
| Lists consume more memory | Tuple consumes less memory as compared to the list |
| Lists have several built-in methods | Tuple does not have many built-in methods. |
| The unexpected changes and errors are more likely to occur | In tuple, it is hard to take place. |

Q41. How can you create a tuple in Python?

A tuple is created by placing all the items (elements) inside parentheses (), separated by commas. A tuple can have any number of items and they may be of different types (integer, float, list, string, etc.).

Q42. Create a tuple and try to add your name in the tuple. Are you able to do it? Support your answer with reason.

my\_tuple = ()

my\_tuple= ("Beci")

print(my\_tuple)

Q43. Can two tuple be appended. If yes, write a code for it. If not, why?

We can't add elements to a tuple because of their immutable property

Q44. Take a tuple as an input and print the count of elements in it.

print(len(my\_tuple))

Q45. What are sets in Python?

A Set is an unordered collection of data items that are unique, means that contains no duplicate elements.

Q46. How can you create a set?

A set is created by placing all the items (elements) inside curly braces {} , separated by comma, or by using the built-in set() function. It can have any number of items and they may be of different types (integer, float, tuple, string etc.).

my\_set = {1, 2, 3}

Q47. Create a set and add "iNeuron" in your set.

new\_set =set()

new\_set.add("iNeuron")

print(new\_set)

Q48. Try to add multiple values using add() function.

new\_set =set()

new\_set.add("iNeuron")

new\_set.add("Data")

print(new\_set)

Q49. How is update() different from add()?

Add() function accepts an element as an argument and if that element is not already present in the set, then it adds that to the set. It returns nothing.

Update(0 function expects a single or multiple iterable sequences as arguments and appends all the elements in these iterable sequences to the set. It returns nothing

Q50. What is clear() in sets?

Clear(0 method removes all items from the set.

Q51. What is frozen set?

The word frozenset indicates that a python set has been frozen. This means the elements of a set which is frozen can't be changed, they are immutable. A frozenset is essentially the unchangeable or immutable version of the set object.

letters=(‘i’,’n’,’u’,’e’,’r’,’o’,’n’)

fSet = frozenset(letters)

Q52. How is frozen set different from set?

**Python set()**

A set is an unordered and unindexed collection of unique elements. Sets are mutable, we can change the elements using a built-in function like add(), remove(), etc. Since the elements are mutable and not in order, they don’t have hash values. So you can’t access the elements with the help of index numbers. Sets can’t be used as a dictionary key or as elements of another set. They can be used as a dictionary value.

**Python frozenset()**

A frozenset is an unordered and unindexed collection of unique elements. It is immutable and it is hashable. It is also called an immutable set. Since the elements are fixed, unlike sets you can't add or remove elements from the set. Frozensets are hashable, you can use the elements as a dictionary key or as an element from another set.

Q53. What is union() in sets? Explain via code.

The Python set union() method returns a new set with distinct elements from all the sets.

s1 = {'Python', 'Java'}

s2 = {'C#', 'Java'}

s = s1.union(s2)

print(s)

Q54. What is intersection() in sets? Explain via code.

The intersection() method returns a new set with elements that are common to all sets.

s1 = {'Python', 'Java'}

s2 = {'C#', 'Java'}

s = s1.intersection(s2)

print(s)

Q55. What is dictionary ibn Python?

A dictionary is as simple as placing items inside curly braces {} separated by commas. An item has a key and a corresponding value that is expressed as a pair (key: value). While the values can be of any data type and can repeat, keys must be of immutable type (string, number or tuple with immutable elements) and must be unique.

my\_dict = {}

my\_dict = {1: 'apple', 2: 'ball'}

Q56. How is dictionary different from all other data structures.

A dictionary is a set of key-value pairs, with the keys being unique within the dictionary. This makes the dictionary useful for storing and retrieving values using that unique key.

Q57. How can we delare a dictionary in Python?

To create an empty dictionary, first create a variable name which will be the name of the dictionary.

Then, assign the variable to an empty set of curly braces, {}.

my\_dict = {}

Q58. What will the output of the following?

```

var = {}

print(type(var))

``

<class 'dict'>

Q59. How can we add an element in a dictionary?

There is no add() , append() , or insert() method you can use to add an item to a dictionary in Python. Instead, you add an item to a dictionary by inserting a new index key into the dictionary, then assigning it a particular value.

course["ineuron"] = 10

Q60. Create a dictionary and access all the values in that dictionary.

year = {'Python': 1993, 'JavaScript': 1995, 'HTML': 1993}

print(year.items())

Q61. Create a nested dictionary and access all the element in the inner dictionary.

D = dict(emp1 = {'name': 'Bob', 'job': 'Mgr'},

emp2 = {'name': 'Kim', 'job': 'Dev'},

emp3 = {'name': 'Sam', 'job': 'Dev'})

print(D)

Q62. What is the use of get() function?

get() method is used in Python to retrieve a value from a dictionary. dict. get() returns None by default if the key you specify cannot be found.

Q63. What is the use of items() function?

items() method is used to return the list with all dictionary keys with values. Parameters: This method takes no parameters. Returns: A view object that displays a list of a given dictionary's (key, value) tuple pair.

Q64. What is the use of pop() function?

The pop() method removes the item at the given index from the list and returns the removed item.

Q65. What is the use of popitems() function?

The popitem() method removes the item that was last inserted into the dictionary.

Q66. What is the use of keys() function?

The keys() method extracts the keys of the dictionary and returns the list of keys as a view object.

Q67. What is the use of values() function?

The values() method returns a view object that displays a list of all the values in the dictionary.

Q68. What are loops in Python?

Looping means repeating something over and over until a particular condition is satisfied. A for loop in Python is a control flow statement that is used to repeatedly execute a group of statements as long as the condition is satisfied. Such a type of statement is also known as an iterative statement

Q69. How many type of loop are there in Python?

**Loop Types**

while loop.

for loop.

nested loops.

Q70. What is the difference between for and while loops?

A for loop is a control flow statement that executes code for a predefined number of iterations. The keyword used in this control flow statement is “**for**”. When the number of iterations is already known, the for loop is used.

A loop that runs a single statement or a set of statements for a given true condition. This loop is represented by the keyword "while." When the number of iterations is unknown, a "while" loop is used.

Q71. What is the use of continue statement?

A continue statement ends the current iteration of a loop. Program control is passed from the continue statement to the end of the loop body. A continue statement can only appear within the body of an iterative statement, such as do , for , or while

Q72. What is the use of break statement?

The break statement is frequently used to terminate the processing of a particular case within a switch statement.

Q73. What is the use of pass statement?

The pass statement is used as a placeholder for future code. When the pass statement is executed, nothing happens, but you avoid getting an error when empty code is not allowed. Empty code is not allowed in loops, function definitions, class definitions, or in if statements.

Q74. What is the use of range() function?

The range() function returns a sequence of numbers, starting from 0 by default, and increments by 1 (by default), and stops before a specified number.

Q75. How can you loop over a dictionary?

We can iterate through a Python dictionary using the keys(), items(), and values() methods. keys() returns an iterable list of dictionary keys. items() returns the key-value pairs in a dictionary. values() returns the dictionary values.

**### Coding problems**

Q76. Write a Python program to find the factorial of a given number.

num = int(input("Enter a number: "))

factorial = 1

if num < 0:

print(" Factorial does not exist for negative numbers")

elif num == 0:

print("The factorial of 0 is 1")

else:

for i in range(1,num + 1):

factorial = factorial\*i

print("The factorial of",num,"is",factorial)

Q77. Write a Python program to calculate the simple interest. Formula to calculate simple interest is SI = (P\*R\*T)/100

def simple\_interest(p,t,r):

print('The principal is', p)

print('The time period is', t)

print('The rate of interest is',r)

si = (p \* t \* r)/100

print('The Simple Interest is', si)

return si

simpleinterest(8, 6, 8)

Q78. Write a Python program to calculate the compound interest. Formula of compound interest is A = P(1+ R/100)^t.

def compound\_interest(p,t,r):

print('The principal is', p)

print('The time period is', t)

A = p \* (pow((1 + r / 100), t))

CI = A - p

print("Compound interest is", CI)

return CI

compound\_interest(8, 6, 8)

Q79. Write a Python program to check if a number is prime or not.

num = int(input("Enter a number: "))

# If given number is greater than 1

if num > 1:

for i in range(2, int(num/2)+1):

if (num % i) == 0:

print(num, "is not a prime number")

break

else:

print(num, "is a prime number")

else:

print(num, "is not a prime number")

Q80. Write a Python program to check Armstrong Number.

num = int(input("Enter a number: "))

sum = 0

temp = num

while temp > 0:

digit = temp % 10

sum += digit \*\* 3

temp //= 10

if num == sum:

print(num,"is an Armstrong number")

else:

print(num,"is not an Armstrong number")

Q81. Write a Python program to find the n-th Fibonacci Number.

def Fibonacci(n):

if n<= 0:

print("Incorrect input")

# First Fibonacci number is 0

elif n == 1:

return 0

# Second Fibonacci number is 1

elif n == 2:

return 1

else:

return Fibonacci(n-1)+Fibonacci(n-2)

print(Fibonacci(10))

Q82. Write a Python program to interchange the first and last element in a list.

def interchange(lst):

lst0=lst[0]

lst4=lst[4]

lst[0]=lst4

lst[4]=lst0

return lst

print(interchange([1,2,3,4,5]))

Q83. Write a Python program to swap two elements in a list.

def swapList(sl,pos1,pos2):

n = len(sl)

# Swapping

temp = sl[pos1]

sl[pos1] = sl[pos2]

sl[pos2] = temp

return sl

l= [10, 14, 5, 9, 56, 12]

print(l)

print("Swapped list: ",swapList(l,2,5))

Q84. Write a Python program to find N largest element from a list.

def N\_max\_elements(li):

li.sort()

a=len(li)

return(li[a-1])

print(N\_max\_elements([10, 14, 5, 9, 56, 12]))

Q85. Write a Python program to find cumulative sum of a list.

def cumSum(s):

sm=0

cum\_list=[]

for i in s:

sm=sm+i

cum\_list.append(sm)

return cum\_list

a=[10,20,30,40,50]

print(cumSum(a))

Q86. Write a Python program to check if a string is palindrome or not.

def string\_palindrome(my\_string):

if(my\_string==my\_string[::-1]):

res= "The string is a palindrome"

else:

res= "The string isn't a palindrome"

return res

my\_string=input("Enter string:")

string\_palindrome(my\_string)

Q87. Write a Python program to remove i'th element from a string.

def remove\_string(str1,i):

print("original: ",string)

# Initialise new string

new\_s= ""

# Removing character

for k in range(len(string)):

if k!=i-1:

new\_s=new\_s+string[k]

return(new\_s)

str="Python"

pos=1

remove\_string(str,i)

Q88. Write a Python program to check if a substring is present in a given string.

def check(str1, sstr):

if (str1.find(sstr) == -1):

print(sstr,"IS NOT PRESENT IN THE GIVEN STRING")

else:

print(sstr,"IS PRESENT IN THE GIVEN STRING")

str1 = input("Enter the string ::>")

sstr=input("Enter Substring ::>")

check(str1, sstr)

Q89. Write a Python program to find words which are greater than given length k.

def word\_k(k, s):

word = s.split(" ")

for x in word:

if len(x)>k:

print(x)

k = 3

s ="Python is good"

word\_k(k, s)

Q90. Write a Python program to extract unquire dictionary values.

test\_dict = {'gfg': [5, 6, 7, 8],

'is': [10, 11, 7, 5],

'best': [6, 12, 10, 8],

'for': [1, 2, 5]}

print("The original dictionary is : " , test\_dict)

res = list(sorted({ele for val in test\_dict.values() for ele in val}))

print("The unique values list is : " , res)

Q91. Write a Python program to merge two dictionary.

dict\_1 = {1: 'a', 2: 'b'}

dict\_2 = {2: 'c', 4: 'd'}

dict\_3 = dict\_2.copy()

dict\_3.update(dict\_1)

print(dict\_3)

Q92. Write a Python program to convert a list of tuples into dictionary.

```

Input : [('Sachin', 10), ('MSD', 7), ('Kohli', 18), ('Rohit', 45)]

Output : {'Sachin': 10, 'MSD': 7, 'Kohli': 18, 'Rohit': 45}

```

Q93. Write a Python program to create a list of tuples from given list having number and its cube in each tuple.

```

Input: list = [9, 5, 6]

Output: [(9, 729), (5, 125), (6, 216)]

```

Q94. Write a Python program to get all combinations of 2 tuples.

```

Input : test\_tuple1 = (7, 2), test\_tuple2 = (7, 8)

Output : [(7, 7), (7, 8), (2, 7), (2, 8), (7, 7), (7, 2), (8, 7), (8, 2)]

```

Q95. Write a Python program to sort a list of tuples by second item.

```

Input : [('for', 24), ('Geeks', 8), ('Geeks', 30)]

Output : [('Geeks', 8), ('for', 24), ('Geeks', 30)]

```

Q96. Write a python program to print below pattern.

```

\*

\* \*

\* \* \*

\* \* \* \*

\* \* \* \* \*

```

for i in range(5):

for j in range(i+1):

print("\* ", end="")

print("\n")

Q97. Write a python program to print below pattern.

```

\*

\*\*

\*\*\*

\*\*\*\*

\*\*\*\*\*

```

def pypart2(n):

k = 2\*n - 2

for i in range(0, n):

for j in range(0, k):

print(end=" ")

k = k - 2

for j in range(0, i+1):

print("\* ", end="")

print("\r")

n = 5

pypart2(n)

Q98. Write a python program to print below pattern.

```

\*

\* \*

\* \* \*

\* \* \* \*

\* \* \* \* \*

```

k = 0

rows=5

for i in range(1, rows+1):

for space in range(1, (rows-i)+1):

print(end=" ")

while k!=(2\*i-1):

print("\* ", end="")

k += 1

k = 0

print()

Q99. Write a python program to print below pattern.

```

1

1 2

1 2 3

1 2 3 4

1 2 3 4 5

```

for i in range(5):

for j in range(i+1):

print(j+1, end=" ")

print("\n")

Q100. Write a python program to print below pattern.

```

A

B B

C C C

D D D D

E E E E E

```

ascii\_value = 65

for i in range(rows):

for j in range(i+1):

alphabet = chr(ascii\_value)

print(alphabet, end=" ")

ascii\_value += 1

print("\n")